ARO 24912.10-EL

SRI International

Final Report • August 1991

ALGORITHMS FOR RADIO NETWORKS WITH DYNAMIC TOPOLOGY

Nachum Shacham, Associate Center Director
Richard G. Ogier, Senior Research Engineer
Vladislav V. Rutenburg, Research Engineer
Information and Telecommunications Sciences Center

Jose Garcia-Luna-Aceves, Center Director
Network Information Systems Center

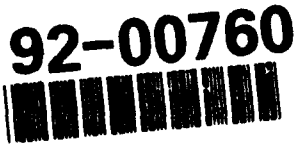
SRI Project 6229

ITAD-6229-FR-91-103

Prepared for:

Department of the Army
U.S. Army Laboratory Command
Army Research Office
P.O. Box 12211
Research Triangle Park, North Carolina 22709-2211

Attention: Dr. James Gault, Technical Monitor



[illegible] • Menlo Park, CA 94025-3493 • (415) 326-6200 • FAX: (415) 326-5512 • Telex: 334486

Final Report • August 1991

ALGORITHMS FOR RADIO NETWORKS WITH DYNAMIC TOPOLOGY

Nachum Shacham, Associate Center Director
Richard G. Ogier, Senior Research Engineer
Vladislav V. Rutenburg, Research Engineer
Information and Telecommunications Sciences Center

Jose Garcia-Luna-Aceves, Center Director
Network Information Systems Center

SRI Project 6229

ITAD-6229-FR-91-103

Prepared for:

Department of the Army
U.S. Army Laboratory Command
Army Research Office
P.O. Box 12211
Research Triangle Park, North Carolina 22709-2211

Attention: Dr. James Gault, Technical Monitor

Approved:

Boyd C. Fair, Director
Information and Telecommunications Sciences Center

Michael S. Frankel, Vice President and Director
Information, Telecommunications, and Automation Division

SRI International 333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326-6200 • FAX (415) 326-5512 • Telex 334486

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1991	Final Report 15 May 88 - 14 May 91

4. TITLE AND SUBTITLE

Algorithms for Radio Networks With Dynamic Topology

5. FUNDING NUMBERS

DAAL03-88-K-0054

6. AUTHOR(S)

Nachum Shacham Vladislav V. Rutenburg
Richard Ogier Jose Garcia-Luna-Aceves

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

8. PERFORMING ORGANIZATION REPORT NUMBER

ITAD-6229-FR-91-103

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARO 24912.10-EC

11. SUPPLEMENTARY NOTES

The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

SRI is pleased to submit this Final Report for the project entitled "Algorithms for Radio Networks with Dynamic Topology," supported by the U.S. Army Research Office under contract DAAL03-88-K-0054. This report presents a brief summary of work performed for this project. Detailed presentations of our results are contained in our publications, which are cited in the report.

The objective of this project was the development of advanced algorithms and protocols that efficiently use network resources to provide optimal or nearly optimal performance in future communication networks with highly dynamic topologies and subject to frequent link failures. As reflected by this report, we have achieved our objective and have significantly advanced the state of the art in this area.

The research topics of the papers summarized include the following: efficient distributed algorithms for computing shortest pairs of disjoint paths; minimum-expected-delay alternate routing algorithms for highly dynamic unreliable networks; algorithms for loop-free routing; multipoint communication by hierarchically encoded data; efficient algorithms for extracting the maximum information from event-driven topology updates; methods for the neural network solution of link scheduling and other difficult problems arising in communication networks; and methods for robust routing in networks subject to sophisticated attacks.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

1 INTRODUCTION

SRI is pleased to submit this Final Report for the project entitled "Algorithms for Radio Networks with Dynamic Topology," supported by the U.S. Army Research Office under Contract DAAL03-88-K-0054. This report presents a brief summary of work performed for this project. Detailed presentations of our results are contained in our publications, which are cited below.

The objective of this project was the development of advanced algorithms and protocols that efficiently use network resources to provide optimal or nearly optimal performance in future C^3I communication networks. These networks will have the following important properties:

- Highly dynamic topology with frequent link failures, requiring algorithms that respond quickly and provide alternate paths;
- Limited communication bandwidth, requiring efficient algorithms;
- Users with diverse requirements, requiring algorithms that support these requirements;
- Multiple transmission media, requiring algorithms that support these media;
- Sophisticated adversary, requiring methods that are robust to intelligent jamming.

As reflected by this summary, we have achieved our objective and have significantly advanced the state of the art in this area.

The summary of our work is organized as follows: In Section 2, we summarize our papers [1, 2, 3, 4], which present new efficient distributed algorithms for computing shortest pairs of disjoint paths. In Section 3, we summarize one of our papers [5], which presents new minimum-expected-delay alternate routing algorithms for highly dynamic unreliable networks. In Section 4, we summarize another paper [6], which presents new routing algorithms for minimum-expected-cost routing in internets with packet loss, and presents a new method for fair charging of network usage. In Section 5, we summarize three papers [7, 8, 9], which present new algorithms for loop-free routing. In Section 6, we summarize our paper [10], which presents a new method for multipoint communication that uses hierarchically encoded data to provide different users with signals of different resolution, rather than exclude some users entirely. In Section 7, we summarize our book chapter [11], which presents an overview of stochastic models for multihop packet radio networks. In Section 8, we summarize our paper [12], which presents new methods for efficiently extracting the maximum information from event-driven topology updates. In Section 9, we summarize two papers [13, 14], which present new methods for the neural network solution of link scheduling and other difficult problems arising in communication networks. In Section 10, we summarize our last paper [15], which presents new methods for robust routing in networks subject to sophisticated attacks. Conclusions are given in Section 11, and a list of references appears at the end of this report.

2 EFFICIENT DISTRIBUTED ALGORITHMS FOR COMPUTING SHORTEST PAIRS OF DISJOINT PATHS

One of the most important current directions in the area of routing is that of reducing network vulnerability. An effective method for reducing this vulnerability is to provide each source with more than one path to its destination. For example, this approach is highly effective when communication interruptions and delays must be minimized while the network is experiencing topological changes or dynamic loading conditions. When links are likely to fail, the source can send duplicate packets, one on each path, to increase delivery reliability. A source node can also monitor the end-to-end performance of the two paths available to it, and send its data on the best one. In addition, if the source requires a higher throughput than a single path can provide, it can split its traffic between the two paths. When the paths are *disjoint*, the greatest benefits result: the reliability is maximized, the available throughput increases, and the traffic conditions along the two paths are least likely to be correlated.

In the papers [1, 2, 3, 4], we present new efficient distributed algorithms for computing disjoint paths. This section summarizes the results of those papers and compares our results to previous work.

Let $G = (V, E)$ be a directed graph containing a finite set of nodes $i \in V$, one of which is a distinguished *destination* node z, and a set of directed links $(i, j) \in E$, each with a nonnegative *length* $c(i, j)$. The number of nodes and number of links will be denoted $|V|$ and $|E|$, respectively. The length of a path P is denoted $c(P)$ and is defined to be the sum of its link lengths. We assume that G is connected and has no zero-length cycles, i.e., that $c(C) > 0$ for any cycle C in G.

We consider the following *shortest pairs of disjoint paths problem* (SPDP): For each node $i \neq z$, find a pair of link-disjoint (or node-disjoint) paths from i to z of minimum total length. (Thus we seek $|V| - 1$ pairs of paths.)

We present distributed synchronous and asynchronous algorithms for both the link-disjoint and node-disjoint versions of SPDP. The algorithms present the nodes with sufficient information to allow the immediate forwarding of packets along the shortest pair of disjoint paths from each node i to z, if such a pair exists.

In [1, 2], we show that SPDP can be reduced to the problem of finding a "minimal" shortest path from each node to the destination in a modified network G', and presents a distributed algorithm on the original network that simulates a shortest-paths algorithm running on the modified network.

The space complexity of the new distributed algorithm is $O([\Delta_i + D] \log |V|)$ for each node i, where Δ_i is the number of neighbors of node i and D is the depth of a shortest-path spanning tree directed toward z. This is because each node must know the identity of each neighbor and each node on its shortest path. This space complexity is better than that of any previous distributed algorithm for the same problem, as discussed below.

We also present an efficient method for forwarding packets on the disjoint paths that does not require any space in addition to that required by the algorithm, and that requires at most one node identity in addition to the destination's identity to be included in each packet. This implies, for example, that (a) source routing (in which the entire path is included in each packet) is not required; and (b) a separate routing table entry for each source (which would

result in a space complexity of $O(|V| \log |V|))$ is not required. The problem of forwarding packets without using source routing or requiring a table entry for each source is nontrivial.

A synchronous implementation of the algorithm is shown to have communication complexity $O(|E||V| + |V|^2 D)$ and time complexity $O(|V|D)$, which are better than for any previous distributed algorithm for the same problem except the "full-information" algorithm that requires each node to know the entire network topology (discussed below).

In [3], we present improvements and extensions to the algorithms of [2] for the shortest pairs of disjoint paths problem (SPDP). In particular, we present synchronous and asynchronous distributed algorithms for SPDP that have better communication complexities and, under certain restrictions on the link lengths, better time and space complexities, than the algorithms of [2].

Assuming all link lengths to be positive integers, we present a synchronous algorithm having communication complexity $O(|E| \log D + |V|D)$ and time complexity $O(D_2 W) = O(|V|W)$, where W is the maximum link length, D is the depth of a shortest-path spanning tree directed toward the destination, and D_2 is the maximum, over all nodes i, of the minimum, over all pairs of link-disjoint (or node-disjoint, depending on the version) paths from i to the destination, of the total number of links in the pair. It's efficiency is due in part to a novel timing of messages. For the important case $W = O(1)$, the time complexity becomes $O(D_2) = O(|V|)$. In this case, the asynchronous algorithm obtained by applying Synchronizer α of Awerbuch [16] has communication complexity $O(|E|D_2)$ and time complexity $O(D_2)$. Two other synchronizers of Awerbuch can be used to obtain a tradeoff between the two complexities.

For the special case $W = 1$, we show that we can drop the requirement that each node know its shortest path in exchange for increasing the communication complexity of the synchronous algorithm to $O(|E|D)$. The space complexity is thus reduced to $O(\Delta_i \log |V|)$ for each node (equivalently, $O(\log |V|)$ per incident link)

Using the theory of biconnectivity, we also show how to extend the above algorithms to efficiently compute shortest pairs of *maximally* link- or node-disjoint paths to the destination when a disjoint pair does not exist. We prove that these new algorithms have the same communication complexity as the corresponding versions of the algorithms for SPDP. The time complexities are also the same, assuming the prior knowledge of the biconnected components of G, which can be computed with $O(|E|)$ communication complexity and $O(|V|)$ time complexity, using the asynchronous depth-first-search algorithm of [17].

Comparison to Previous Work

A distributed algorithm for computing disjoint paths was developed by Itai and Rodeh [18]. That algorithm, whose communication and time complexities are both $O(|V|^2)$, does not attempt to find the shortest pair and can result in very long paths. A different approach for finding disjoint paths was proposed for the ARPANET [19], and is based on first obtaining the shortest path between the source and the destination, then increasing the costs of the path's links and recomputing the shortest path. It is easy to see that this approach can fail to find a pair of disjoint paths even if one exists.

A simple algorithm for computing a shortest pair of disjoint paths from a single source i to a single destination z is described by Suurballe [20]. That algorithm uses a network flow approach and involves first finding the shortest path from i to z, and then finding the shortest "augmenting" path. The synchronous distributed Bellman-Ford shortest-paths algorithm (e.g., [21]) can be used to compute this path with communication complexity $O(|E||V|)$, time complexity $O(|V|)$, and space complexity $O(\Delta_i \log |V|)$ for each node i. SPDP can therefore be solved by running this algorithm for every source i, resulting in communication complexity $O(|E||V|^2)$, time complexity $O(|V|^2)$, and space complexity $O(|V|\Delta_i \log |V|)$ for each node i. This compares to $O(|E||V| + |V|^2 D)$, $O(|V|D)$, and $O([\Delta_i + D] \log |V|)$ for the synchronous algorithm presented in [2].

The best previously known synchronous or asynchronous algorithm for SPDP is the "full-information" algorithm (e.g., [22]), in which the entire network topology is delivered to each node, and each node solves the problem using a centralized algorithm. The full-information algorithm has communication complexity $O(|E||V|)$ and time complexity $O(|E|)$. Our synchronous algorithm of [2] is better in time complexity and equal in communication complexity if $|V|D < E$ (which is the case for dense networks). In addition, the full-information algorithm has a space complexity of $O(|E| \log |V|)$, which is much worse than for our new algorithm. Our asynchronous algorithm of [3] has communication complexity $O(|E|D_2)$ and time complexity $O(D_2)$, assuming $W = O(1)$, which beats the full-information algorithm in both complexities.

3 MINIMUM-EXPECTED-DELAY ALTERNATE ROUTING

In the paper [5], we consider the important problem of minimum-expected-delay routing in an unreliable network subject to frequent random link failures. Such link failures can represent jamming, natural interference, or simply the unavailability of a link for which another user currently has priority.

Previous routing algorithms are inadequate in several ways when applied to highly dynamic networks. Shortest-path algorithms must recompute a new shortest path whenever a link fails, requiring unnecessary time and communication overhead, especially if link failures are frequent. As discussed below, alternate-routing algorithms based on shortest paths fail to minimize expected delay because they do not take into account the existence of alternate routes at downstream nodes; in this sense they are myopic.

We assume that the network has an underlying topology given by a directed graph $G = (V, E)$, where E represents the set of links that are potentially operational. Each link in E switches according to a Markov chain between two possible states: up and down. The Markov chains for different links are independent. To simplify the presentation, we assume that time is divided into slots of equal length. We let $\lambda(i,j)$ denote the probability that link (i,j) will come up in the next slot given that it is down in the current slot, and we let $\mu(i,j)$ denote the probability that link (i,j) will go down in the next slot given that it is up in the current slot. It follows that the steady-state probability that link (i,j) is up is

$$p(i,j) = \frac{\lambda(i,j)}{\lambda(i,j) + \mu(i,j)} \quad . \tag{3.1}$$

(If $\lambda(i,j)$ and $\mu(i,j)$ are both zero, then $p(i,j)$ can be any number between 0 and 1.) Link (i,j) has a delay of $d(i,j)$ slots when it is up, which can represent propagation and queuing delays. We assume G, $d(i,j)$, $\lambda(i,j)$, and $\mu(i,j)$ are quasi-static, i.e., change slowly. We emphasize, however, that the actual state of each link can change arbitrarily frequently.

We assume that each node i knows the current state of its outgoing links (i,j). It is clearly unreasonable to assume that each node knows the current state of every link in the network, since such information is likely to be outdated by the time it is received. Therefore, we make the more reasonable assumption that each node knows the above quasi-static information for every link in the network. (In practice, this information can be broadcast periodically.)

We consider the following *minimum-expected-delay routing problem*: find a routing strategy that minimizes, over all possible routing strategies, the expected time for a packet originating from any node to reach a given destination z.

This problem is not equivalent to finding a shortest path from each source to z, since the links of any single path can change their state while the packet is en route. Instead, the routing decision at each node of the packet's journey must depend on the current state of the node's outgoing links. This problem is also not equivalent to finding k shortest paths with distinct initial links from each source to z [23]. Although the solution to the latter problem provides alternate paths that can be used in case an initial link fails, it does not consider the existence of alternate paths that can be taken in case a noninitial link of the path fails, potentially resulting in a much higher delay.

We emphasize that the problem we consider is to find the minimum-expected-delay routing strategy among *all* possible strategies, including those that are history dependent (in which the decision at each hop depends on where the packet has traveled so far) and those that contain loops. To see that the optimal routing strategy can contain a loop, consider a simple network with nodes 1, 2, and z and links $(1,z)$, $(2,z)$, $(1,2)$, and $(2,1)$. In this example, $d(i,j) = 1$ for all links, links $(1,2)$ and $(2,1)$ are always up, $p(1,z) = p(2,z) = .5$, and $\lambda(1,z) = \lambda(2,z) = .1$. Thus, if link $(1,z)$ or $(2,z)$ is down, it will stay down for an average of 10 slots. Clearly, if both of these links are down, then the minimum-expected-delay routing strategy is for a packet to hop between nodes 1 and 2 until one of the down links comes up.

The general minimum-expected-delay routing problem as stated is very difficult. In fact, we show that it is #P-complete (the enumeration version of NP-complete) by transformation from the two-terminal reliability problem, which was shown by Valiant [24] to be #P-complete. We therefore cannot expect to solve the general problem in polynomial time, and so we consider the following two more tractable cases:

Case 1 G is a DAG (directed acyclic graph) oriented toward z. Equivalently, G contains no cycles and z is the only node with no outgoing link.

Case 2 The states of each link are independent from slot to slot. Equivalently, $\lambda(i,j) = p(i,j)$ for all links.

Using the theory of controlled Markov processes, we show that the optimal routing strategy for either case has the following form: Each node i is assigned an ordering on its outgoing links (best to worst) and an integer $K(i)$. We let P_i denote the ordered set containing the $K(i)$ best links at node i. The links of P_i are called *preferred links.* If node i has a packet to forward, and at least one preferred link is up, then the packet is forwarded on the best such link; otherwise node i waits for a preferred link to come up. P_i does not depend on the current state of node i's outgoing links (it only depends on the quasi-static information G, d, λ, and μ).

Since the set P_i provides alternate links that are computed in advance of any link failures, the optimal solution takes the form of alternate routing. We therefore call our solution *minimum-expected-delay alternate routing* (MEDAR). We emphasize that, although P_i does not depend on the the current states of node i's outgoing links, the MEDAR solution is optimal among all routing strategies that do depend on these states.

For the two cases above, we present algorithms with surprisingly low time complexities: $O(|E|)$ for Case 1 and, assuming unit link delays, $O(|E| \log_{(1+|E|/|V|)} |V|)$ for Case 2. These algorithms are centralized but can be run at each node, given the global quasi-static information discussed above. In addition, we present heuristic centralized and distributed algorithms that have been shown by simulation to perform well for the general model.

In summary, MEDAR provides the following practical benefits:

- It makes optimal use of statistics for link-state dynamics to minimize expected delay.
- By computing a quasi-static routing structure that provides alternate paths, it requires less frequent topology updates and avoids the delay of computing a new path if a link fails.

- It allows the option of waiting for an outgoing link to come up rather than immediately using an inferior link.

4 MINIMUM-EXPECTED-COST ROUTING IN INTERNETS WITH PACKET LOSS

In the paper [6], we propose an economics-based definition of fair pricing and derive fair, simple, and mathematically sound charging policies that properly handle costs and risks in lossy internetworks. We also describe simple and efficient source-based algorithms (having both link-state and distance-vector versions) for computing the optimal routing tables with respect to the new charging measures. These charging measures and algorithms have the property that, by minimizing the individual domain's expenses, they also minimize the global resource consumption. All of the above results and algorithms hold when we replace "price" with "delay." Thus, we present efficient shortest-path algorithms for computing minimal expected delay (with retransmission) routing in lossy networks.

A great deal of attention and effort are currently devoted to developing a new generation of protocols that facilitate the routing of packets through multiple administrative (or autonomous) domains (ADs) that make up the rapidly growing Internet ([25], [26], [27], [28]). Our paper presents a rigorous and general approach to one of the most important (and previously unresolved) research issues in internetwork accounting policies: the issue of fair, efficient billing and cost recovery policies in internets and, in particular, the highly difficult aspect of billing and routing in internets comprised of potentially lossy ADs, i.e., ADs that sometimes lose or drop packets. We present fair billing mechanisms for internetwork activities involving cooperation of lossy ADs, and describe efficient algorithms for computing the best packet routing within internetworks.

We assume that the Internet consists of many individual ADs. Each of these ADs is attached to some other ADs through gateways. In the basic mathematical model, we represent the Internet by a graph G, each AD as a node in G, and draw a link between any two nodes that are physically attached to each other through gateways. This model will be used throughout most of this paper unless stated otherwise. This is not the most detailed model, because in reality each AD, represented here as a node, is a large network in itself, consisting of many switching nodes and hosts.

We are concerned with the billing policies for important or necessary packets, i.e., those packets whose loss requires retransmission. We consider the two basic retransmission policies. The first policy, referred to as *route based*, assumes that the source keeps a copy of the packet and retransmits the packet if it gets lost. The second, referred to as *hop based*, assumes that a copy of the packet is kept at each intermediate AD until that AD knows that the packet has successfully cleared the next domain. The AD will retransmit the packet if it gets lost in the next domain. This paper concentrates on the route-based policy because it involves more difficult mathematical issues. However, the corresponding results for the hop-based policy are also presented.

We have developed cost recovery schemes that assign fair wages to individual domains for successful delivery of packets and fair penalties for losing packets. These methods ensure that each domain breaks even in the long run. The new prices (wages) reflect both the delivery costs and the risks of losses in an economically optimal way.

In particular, we define the notion of price fairness. We consider a packet transiting through k administrative domains $N_1, N_2, ..., N_k$ on its path P from the source $s \in N_1$ to the destination $d \in N_k$. Let c_i denote the cost that domain N_i incurs for handling this packet,

and let p_i denote the probability of a packet getting lost inside domain N_i. Let e_i denote the fair tariff that domain N_i should charge to the sender for successfully transitioning a packet, and let u_i denote the penalty that domain N_i should pay to the sender for losing a packet. The fairness considerations imply that the price e_i is fair if it allows region i to break even in the long run; i.e., the expected return Q_i should be equal to 0.

Fair pricing policy should be as follows: If domain N_i loses the packet, it should reimburse the client in the amount u_i equal to E_{i-1}. If domain N_i successfully transitions the packet, it should be paid by the client in the amount of

$$e_i = \frac{p_i E_{i-1} + c_i}{1 - p_i}.$$

Here, $E_{i-1} = \Sigma_{j=1}^{i-1} e_j$ denotes the total cost accumulated by a packet entering domain N_i.

We also present efficient algorithms for finding the least expensive (with respect to the above prices e_i) paths between different pairs of domains in lossy internets.

We derive that, given a fixed path P, the total price J_P for successful packet delivery from source to destination along the path P is given by

$$J_P = \frac{\sum_{i=1}^{k} c_i R_{i-1}}{R_k},$$

where $R_0 = 1$ and $R_i = (1 - p_1)(1 - p_2)(1 - p_3) \cdots (1 - p_{i-1})(1 - p_i)$.

Based on this new price measure, one can now try to find the path that minimizes the value of J_P. Because the price J_P is, in our case, a function of the total accumulated cost, the choice of the future portion of the path depends on its past. Because of this fact, the commonly used, efficient destination-based versions of algorithms (Dijkstra, Bellman-Ford) for computing optimal routing do not work with respect to the new measure. The lack of a dynamic programming approach results in highly burdensome algorithms with exponential time complexity.

The problem of prohibitively high algorithm complexity can be resolved by the following idea: compute optimal routing from a fixed source s to all destinations, rather than from all sources to a fixed destination d. The efficient dynamic programming algorithms are applicable to this approach because the memoryless property holds in the backward direction, i.e., the portion of an optimal path before some intermediate node w is independent of the path after w.

Thus, the equally efficient source-based versions of these algorithms do work, providing an efficient means of computing optimal routing between different source-destination pairs. The most efficient centralized version of this algorithm can be implemented in $O(|E| \log_{2+|E|/|V|} |V|)$ time and $O(|E|)$ space, where $|V|$ is the number of nodes (ADs) in G and $|E|$ is the number of links in G, using the efficient versions of Dijkstra algorithm developed in [29] and [30]. If the network protocols use distance vectors, the best approach is to use reversed (i.e., source-based) versions of the distributed Bellman-Ford algorithm (see [31] and [21]). The synchronous distributed Bellman-Ford algorithm has message complexity $O(|E||V|)$ and time complexity $O(|V|)$.

An important aspect of the results of this paper is that, no matter what the individual billing policies are, this paper's measures and algorithms should still be used in order to minimize the average global resource cost of successful packet delivery. Namely, given a path P in G, the expected total system cost with retransmission until successful delivery is proved to be equal to $(1/R_k)\sum_{i=1}^{k} c_i R_{i-1}$, which is exactly equal to the path cost J_P derived in the previous paragraphs. Thus, the optimal routes with respect to J_P also minimize the global resource use.

In addition, we present a distributed destination-based algorithm for computing optimal paths that allows for faster response time in highly dynamic and unreliable internets. We also present preliminary accounting procedures for implementing the proposed charging policies.

The results of this paper can be summarized as follows:

- We propose a new mathematical definition of fairness and derives fair and simple charging policies that properly handle costs and risks in lossy internetworks.
- We present simple and efficient shortest-path algorithms for computing the optimal routing tables with respect to the new price measures.
- These algorithms also find paths that minimize the global resource consumption over all possible paths and flows.
- The proposed charging policies lead to a general methodology for incorporating non-standard policies of individual administrative domains.
- Preliminary accounting procedures for implementing the billing policies are also presented.

All of the above results and algorithms hold when we replace cost with delay. Thus, we present simple and efficient shortest-path algorithms for computing the optimal routing tables with respect to minimizing the expected delay (with retransmission) until delivery in lossy networks.

5 LOOP-FREE ROUTING ALGORITHMS FOR NETWORKS AND INTERNETS

The routing protocols used in most of today's computer networks are based on shortest-path algorithms that can be classified as *distance-vector* or *link-state* algorithms. In a distance-vector algorithm, a node knows the length of the shortest path from each neighbor node to every network destination, and uses this information to compute the shortest path and next node in the path to each destination. A node sends update messages to its neighbors, who in turn process the messages and send messages of their own, if needed. Each update message contains a vector of one or more entries, each of which specifies, as a minimum, the distance to a given destination. In contrast, in a link-state algorithm, also called a topology-broadcast algorithm, a node must know the entire network topology, or at least receive that information, to compute the shortest path to each network destination. Each node broadcasts update messages, containing the state of each of the node's adjacent links, to every other node in the network.

Several routing protocols based on distance-vector algorithms, called *distance-vector protocols* (DVPs), have been proposed for and implemented in computer networks, including the old ARPANET routing protocol, the stationless routing protocol of the DARPA packet-radio network (CAP 7), and the routing protocols used in Datapac and in the Digital Network Architecture (DNA) Phase IV [7] [32]. Well-known examples of DVPs implemented in internetworks are the Routing Information Protocol (RIP), the HELLO protocol, the Gateway-to-Gateway Protocol (GGP), and the Exterior Gateway Protocol (EGP) [8]. All of these DVPs have used variants of the distributed Bellman-Ford algorithm for shortest-path computation [33]. The primary disadvantages of this algorithm are *routing-table loops* and *counting to infinity* [34] [7]. A routing-table loop is a path specified in the nodes' routing tables at a particular point in time, such that the path visits the same node more than once before reaching the intended destination. A node counts to infinity when it increments its distance to a destination until it reaches a predefined maximum distance value.

A number of attempts have been made to solve the counting-to-infinity and routing-table-looping problems of distance-vector algorithms by increasing the amount of information exchanged among nodes, or by making nodes to hold down the updating of their routing tables for some period of time after detecting distance increases. However, none of these approaches solves these problems satisfactorily [7]. A recent DVP developed for internetwork routing, called the Border Gateway Protocol (BGP) [27], specifies the entire path from source to destination in update messages, as proposed by Shin and Chen [35], to *detect* the occurrence of loops.

On the other hand, link-state algorithms are free of the counting-to-infinity problem. However, they need to maintain an up-to-date version of the entire network topology at every node, which may constitute excessive storage and communication overhead in a large, dynamic network. It is also interesting to note that the routing protocols using link-state algorithms, called link-state protocols (LSP), which have been implemented to date do not eliminate the creation of temporary routing-table loops. Well-known examples of LSPs are the OSI intradomain routing protocol [36], and the Open Shortest Path First (OSPF) protocol [37].

Whether link states or distance vectors are used, the existence of routing-table loops, even temporarily, is a detriment to overall performance of an internet. The two main results of the research carried out in this area are as follows:

- Providing a unified approach to the solution of the counting-to-infinity and routing-table-looping problems in distributed routing algorithms that use either distance vectors or link states [8].

- Unifying new and previous results on loop-free routing using distance vectors into a new distance-vector algorithm, called the diffusing update algorithm (DUAL), that is always loop-free, operates with arbitrary transmission or processing delays, assumes arbitrary positive link costs, and provides shortest paths within a finite time after the occurrence of an arbitrary sequence of link-cost or topological changes [7] [9].

The above results are based on the concept of diffusing computations and new feasibility conditions used to determine when network nodes need not coordinate with other nodes when they change their successors (next hops) to destinations.

In DUAL, when a node needs to update its routing table for a given destination j, after it processes an update message from a neighbor or detects a change in the cost or availability of a link, the node tries to obtain a new *feasible successor* with the shortest distance to node j. From node i, a feasible successor toward node j is a neighbor node that has reported a distance to the destination that is smaller than node i's own distance to the same destination. When feasible successors are found, the algorithm behaves as the distributed Bellman-Ford algorithm. If a node cannot find a feasible successor with the shortest distance to node j, the node starts a diffusing computation [38] for node j by sending a *query* to all its neighbors. The node cannot change its successor to node j until it receives a *reply* for its query from each neighbor; the reply indicates that the neighbor has processed the query and has either obtained alternative feasible successors to node j, or determined that it cannot reach node j. Once node i obtains all the replies to its query, it computes a new distance and successor to destination j and sends an update.

An update, query, or reply simply specifies the distance to a destination. Because contiguous sequences of DUAL updates for the same destination are redundant, only the last update for each contiguous sequence is included in the packet actually sent in the simulation. Multiple changes in link cost or availability are handled by ensuring that a given node is waiting to complete the processing of at most one query at any given time.

DUAL was proven to be free of routing-table loops at every instant, regardless of the type or number of changes in the network [9], and to converge to correct routing-table values within a finite time after the occurrence of an arbitrary sequence of link-cost or topological changes. DUAL has also been shown [9] [39] to match or improve the performance of all previous loop-free distance-vector algorithms [34] [40]. The only concern regarding DUAL's performance is after node failures and network partitions, because in such cases all network nodes have to be involved in the same diffusing computation. Fortunately, hold-downs can be used to improve DUAL's performance. Hold-downs have been used in the past in DVPs [7] unsuccessfully to try to eliminate looping. In DUAL, however, a hold-down is used not to prevent looping but to simply reduce the number of update messages exchanged after a

resource failure; therefore, it provides better or at least the same level of performance than would be obtained if no hold-down were used.

6 MULTIPOINT COMMUNICATION BY HIERARCHICALLY ENCODED DATA

Multicast is a service in which a source sends information to multiple recipients. This service will be an important element in the emerging broadband network technologies that will carry a large variety of traffic types, many of which, including video, and imaging, are of high rate and often require transmission to many recipients [41, 42].

To be able to support multicast services, broadband networks must supplement their increased bandwidth and improved switching facilities by protocols for generation, management, routing, and transport of multicast-oriented signals. Most existing protocols at the transport and network layer, e.g., TCP and IP [43], were designed for narrow-band network environments, and support communication only with a single recipient (unicast).

All of the multicast protocols were developed thus far under the assumption that all recipients must receive all the information emitted by a source [44, 45, 46, 47, 48, 49]. This, however, may not always be feasible or even desirable, especially when the multicast information is broadband. The user population is expected to be heterogeneous, with the set of multicast recipients greatly differing in the communications end devices they use and the network-access bandwidth available to them. Thus, when a source distributes a high bandwidth signal to multiple users, not all of them are willing to receive or are capable of receiving the complete signal. Many users can receive or are content to receive only a subset of the information contained in a multicast signal. An example for such a scenario is when a video signal is distributed to a (potentially large) number of recipients who widely differ in their display devices and the bandwidth available to them. Users with wideband access, high-resolution displays, and powerful processors can receive and process the complete high-resolution color video signal, whereas users with less capable displays or lower bandwidth access, who are capable of receiving only part of the signal, may prefer to receive, say, only black-and-white, low-resolution video to receiving no video at all. Similarly, in voice communication, users may settle for low-rate synthetic speech without speaker recognition when they cannot receive a complete digital speech signal.

Moreover, users may differ in their ability to receive broadband multicast signals, even if they have similar access bandwidth and terminals. In multimedia teleconferencing, users who send and receive multiple streams representing the various media may not be able to obtain the full bandwidth needed to communicate via all these media simultaneously. Consequently, users must make choices regarding which signal they emphasize at any given time. These decisions, which are made by individual users, are likely to change with time, reflecting a user's ability to focus on different media at different times. For example, users may first allocate most of their access bandwidth for video and de-emphasize computer animation; later, as more computational results are presented through animation, users may change their allocations to allow a close examination of that signal, while receiving only a low-resolution, black-and-white video.

Therefore, using existing multicast protocols, which were designed on the assumption that all destinations must receive the same information, raises a serious dilemma: the session must either exclude the more limited users who cannot receive the full signal, or must penalize the more capable users by compressing (and distorting) the signal to fit the least capable users. In the paper [10], we propose an approach for resolving this dilemma by having the

source transmit its full signal and the network deliver to each recipient portions of the signal as determined by bandwidth availability and terminal capability constraints. Since recipients differ in their capabilities, they also differ in the subsets of signal they receive.

Since current protocols are not designed to operate in this mode, new mechanisms and algorithms must be designed to support such a service. The objective of our paper [10] is to discuss the characteristics of such mechanisms and to present approaches for realizing them. This paradigm of multicast with individualized delivery, as described above, requires cooperation between the connection end points (source and destinations) and the network. We focus on the following major areas of this problem:

Signal Representation

The source must properly encode its generated signal so as to permit selection of subsets of the signal for transmission and reception with signal quality at the receiver proportional to the size of the received subset. Such coding algorithms are known as *layered* or *hierarchical coding* and the corresponding signal representation as *layered* or *hierarchical signals.* We review several layered coding techniques and analyze their suitability for multicast service. These techniques include *subband coding*, which is based on partitioning the spectrum of the video signal along its three-dimensional frequency region (horizontally, vertically, and temporally [50]), and *conditional replenishment*, which is utilized to generate a variable bit-rate (VBR) video stream, based on a receiver reconstructing a video signal of constant quality [51].

The major advantage of hierarchical coding is that it enables users to trade quality for bandwidth, which can be done either in closed or open-loop methods. The former is based on feedback messages to the source to change its rate according to the network's ability to carry its signal and the recipient(s) ability to receive it. The latter, which is more appropriate for high-speed networks, is based on sending the full signal by the source and filtering of unnecessary layers at the network.

Routing

The network must use a routing algorithm that finds optimal paths from the source to all destinations, under the constraints of user demands and available network resources. We present an efficient algorithm for determining multiple paths to recipients so that data to each destination is routed on the shortest path that either satisfies the bandwidth requirement of that recipient or has the maximum possible capacity to that destination. The algorithm operates in stages, where in each stage it construct a collection of paths with maximum available bandwidth. It then restricts the paths to those that lead to multicast destinations and resets the available bandwidth of the remaining links to the minimum of their actual bandwidth and next maximum available bandwidth. The nodes where routes from two stages of the algorithm meet are filtering nodes where signal layers are discarded.

It is interesting to note that while traditional multicast routing algorithms have focused on finding an optimal tree (e.g., a minimum spanning tree or a shortest-paths tree) the set of paths for the problem considered here may not always form a tree. That is, under certain

circumstances a network node may be on two paths that carry two different representations of a signal.

Maintaining Data Integrity

Existing protocols for error- and loss-free data delivery are designed to protect the complete data stream and are mostly based on retransmission. In the problem we consider here, different parts of the data should have different levels of error protection, since errors and lost data have different effects, depending on the layer in which they occur. Also, since different destinations receive different parts of a generated signal, the different layers should be separately protected. Finally, it is expected that many applications of the type of multicast we discuss here will require time-constrained delivery of data, thereby excluding retransmission as a data recovery mechanism. We present new techniques for open-loop recovery of lost data based on erasure correction, and discuss ways of providing different protections to different layers of a signal.

Our new techniques are based on an FEC scheme described by Shacham and McKenney [52], which groups the source's packets into L-packet blocks, and adds to each block one or more *parity packets*, where each bit in a parity packet is a function of the corresponding bits in the data packets of the block. We augment the FEC scheme to take into consideration the special features of hierarchical data. This augmentation includes the following:

- Independent coding of each layer
- Different erasure recovery codes or rates are applied to different layers, to match the level of protection to the signal quality deterioration due to packet loss at the various layers.

The aforementioned collection of techniques provides the means for a new multipoint communication paradigm in which the recipients may receive only parts of a transmitted signal, and the received parts can vary from destination to destination. This allows a multicast session to incorporate many recipients who are unable to receive the full signal and who prefer to receive partial information to receiving no information at all.

7 STOCHASTIC MODELS FOR MULTIHOP PACKET RADIO NETWORKS

When a common radio channel serves as a communication medium for a set of geographically distributed packet switches, the resulting system is called a *packet switching radio network* (PRNET). In a traditional packet switching network, such as ARPANET, a link is used exclusively by the two nodes it connects. In PRNET, however, multiple nodes share the channel either by reserving it for specific periods, or by contending for channel access for each packet they wish to transmit. Channel sharing has a significant impact on the structure of both the network nodes and the protocols they use. It also makes it hard to ascertain PRNET performance.

In the book chapter [11], we provide an overview of the analytic models for the behavior and performance of such networks. In the models we consider, a packet radio (PR) node incorporates a digital unit, which handles packet storage, queueing, processing, and all other tasks required for participation in the network protocols; a single transmitter/receiver (transceiver) unit which can either transmit or receive at any given time but not perform both functions simultaneously and cannot receive more than one packet at a time; and an omnidirectional antenna, which causes transmitted packets to propagate to all nodes in radio range of the transmitter.

Concurrently-transmitted packets are the only source of interference that may prevent a PR from receiving a packet. If there is no interference a packet is successfully received. Three types of interference are considered: (1) self-interference, occurring when a packet is directed at a PR which is transmitting at that time, (2) primary interference, occurring when two nodes direct packets to the same receiver, and (3) secondary interference, occurring when a neighbor of a receiving PR transmits a packet to a third PR. The first type is alway destructive; however, with the second and third types, some of the interfering packets can be received depending on such factors as signal modulation (e.g., spread spectrum) and relative power levels.

A PRNET consists of a set of PRs and a radio medium with specific interference rejection properties. Packets are generated by sources, end devices, or gateways from other networks. Each packet carries sufficient information to route it to its destination. We restrict ourselves to PRNETs in which *not all packets can reach their destination in a single transmission* because of limited transmission range. End-to-end routes must thus be established, with the tandem PRs storing and forwarding the packets to their destinations. Such networks are called *multihop* PRNETs.

The PRNET topology is induced by the radio range of its PRs. That is, two PRs are called neighbors if they can receive each other's transmission when there is no interference.

The PRNETs we consider have the following additional characteristics:

- Channel sharing is governed by a random-access protocol.
- Time may be either slotted or unslotted.
- Packets are addressed and are discarded by PRs that receive them but are not addressees.
- Packets are (re)transmitted until they are received correctly.

The intricate interference patterns existing in this unique environment are extremely difficult to capture in an tractable analytic model; thus, no all-encompassing models have been developed so far. Rather, each of the existing models considers only part of the PRNET functions, and relies on various assumptions and approximations.

The models are presented in four categories:

- *Queueing* models consider a single node and represent the environment's effect on the rate at which it serves the packets by an "interference factor." These models, which require slotted time and assume given network topology and routing, result in throughput and average packet delay, as well as conditions for stability of the queues.
- *Spatial* models consider PRNETs with random topologies. Although many assumptions are made for the sake of tractability, these models, which also require slotted time, allow explicit consideration of routing policies, various propagation models, and even nodal mobility.
- *Channel access* models focus on channel transmission activity. These models take as their input the network topology and routing, and consider unslotted time protocols. They allow us to determine the feasible throughput region of the PRNET, and to study the effect of several channel access protocols on PRNET performance.
- *Chain* models for multihop PRNETs in which the nodes are arranged in a linear array. These models assume slotted time and provide results for the network throughput.

These models provide valuable insight into PRNET operation and the effects of its elements. Their main strength is in comparative evaluation; changes in performance resulting from varying system parameters can be identified, and in some cases the optimal values can be determined. One can also compare the effect of routing strategies and channel access protocols on throughput. As such, the numerical results provided by the above models enrich our basic understanding of PRNET operation and the effects of its underlying processes. However, to make those numerical results applicable to the design process of an operational PRNET, advances must be made in two crucial directions:

- Correlate the results of the various models with one another.
- Verify model results with those of operational PRNETs, or more realistic simulation models.

Progress in these two directions will make the analytic models much more acceptable to the PRNET designers community, and will stimulate further development of more comprehensive models.

8 EXTRACTING MAXIMUM INFORMATION FROM EVENT-DRIVEN TOPOLOGY UPDATES

One of the most important areas of current research is that of designing packet routing and flow control algorithms in networks that undergo dynamic changes due to the failure and recovery of links. A major determining factor for the success of such algorithms is the timeliness of link-state updates, which determines the amount of useful information available to each node (or the operation center) regarding the current and future state of various links in the network. For example, dynamic routing algorithms, that base each hop-by-hop routing decision on their probabilistic estimates of the states of the network links and nodes, depend on the knowledge of the state of all downstream links in the network. These probabilistic estimates depend on the knowledge of the state of each link at some time t in the past and the conditional distribution of the current link state, conditioned on its state at time t. The more recent t is, the better the performance of the routing algorithms.

In the paper [12], we address the issue of designing the best method for link-state updating in communication networks undergoing topological changes. In particular, we conclude that the best way of disseminating link-state information is through event-driven (ED) updating and show how to overcome its main drawback that, under ED updating, one cannot tell whether a lack of new updates from a distant link is caused by the state of that link not changing or by link failures along the updating paths. We present an efficient algorithm that extracts the maximum implicit information from ED topology updates by computing the latest time for which each processing node can be certain of the state of each network link. We show that this algorithm provides the same information as continuous flooding of link-state information (i.e., periodic updating with an infinitesimal period), and yet requires the much lower communication overhead of ED updating.

We also present a method that allows each node to continuously refresh the above information without having to run the above algorithm more than once for each received update. In addition, we present two approximate versions of the above algorithm that reduce communication overhead by limiting the radii of propagation.

In a network (a directed graph) G, link-state updates are usually distributed by flooding over some spanning subgraph H of G. Throughout this discussion, we assume that the update packets are assigned the highest routing priority, and that update packets are lost only because of link failures. The two standard approaches to the timing of updates are *periodic updating*, which is self-explanatory, and ED updating, which involves generating updates in response to changes in link state. The use of periodic updating with long inter-update periods leads to the degradation in the "freshness" of update information. One possible way to ensure the best possible freshness of updates (within the limitations imposed by the dynamic state of the network) is by using continuous flooding (CF) of link-state information (i.e., periodic updating with an infinitesimally short period). However, this approach is extremely wasteful and cannot be implemented in practice.

The above considerations point out the desirability of the ED approach, which is clearly much more efficient and, in fact, very practical. However, we have to answer the question as to how much will the use of ED updating degrade the freshness of link-state information, as compared with the idealized CF paradigm. If only a single link is undergoing dynamic changes in the network, it is not difficult to see that the freshness of information about this

link under the ED approach will be exactly the same as under the CF approach. However, in general, when all of the links are simultaneously experiencing dynamic changes, the straightforward ED approach provides less information than CF. This is because under the ED approach, if a node has not received information about some link for some time, it does not know how recently the state of that link could have changed, since the absence of any new updates could result either from the state of that link not changing, or from a new update being blocked by a disconnected network or delayed by a long path.

We present an efficient algorithm that extracts *implicit* information from ED updates and computes the latest time for which each node can be certain of the state of each link. Intuitively, this algorithm uses the available information about the state of the links in the network to deduce the real cause for the absence of updates from network links. More precisely, the algorithm allows each node i to compute for every link (j, k) in the network the last time T_{jk} for which i is certain of the state of this link, assuming link-state updates are disseminated using ED flooding. This time, which is usually much more recent than the time contained in the latest update received from link (j, k), can be used by node i to estimate the current and future states of distant links. In addition, we present algorithms that compute T_{jk} approximately while reducing the radius of propagation of link-state updates, thus further reducing communication overhead.

9 NEURAL NETWORK SOLUTION TO THE LINK SCHEDULING PROBLEM

To rapidly solve the computationally intensive problems posed by future communication networks, new approaches are needed. These networks will be large (thousands of nodes), dynamic (due to mobility, traffic fluctuations, and node or link failures), very high speed, and heterogeneous, and will support a variety of integrated services (e.g., voice, data, and graphics). The protocols needed to manage these networks effectively must quickly solve difficult, large-dimensional optimization problems, such as type-of-service routing, topology control, link scheduling, and dynamic routing, many of which are NP-complete. Except for a few special cases, no algorithms exist that can compute solutions quickly enough using conventional computers.

In the papers [13, 14], we investigate the applicability of neural network methods to future communication networks. Although neural networks do not always compute exactly optimal solutions, for many problems they quickly compute nearly optimal solutions [53, 54], which are often far more beneficial than slowly computed optimal solutions; this is especially true in highly dynamic communication networks. Moreover, because neural networks can contain millions of processors, they can solve extremely complex problems involving millions of variables, and therefore do not require the unrealistic simplifications or approximations that are often necessary when conventional computers are used to solve difficult communication network problems.

In a commonly used neural network method for solving optimization problems [53], the neural network uses a form of gradient descent to find a local minimum of an "energy function" that is determined by the network's connection strengths and input currents. The energy function is chosen such that its global minimum solves the optimization problem. A problem with this method is that the energy function is usually nonconvex, and the local minimum found by the neural network often represents a solution that is far from optimal. In the papers [13, 14], we have developed two new methods that overcome this problem.

In the paper [13], we present a new method, called *convex relaxation*, for the neural network solution of combinatorial optimization problems, and apply this method to the NP-complete problem of finding a minimum-length link schedule in a multihop radio network with given link demands. In the convex relaxation method, the energy function associated with the neural network is initially convex, and is gradually modified until it becomes the nonconvex energy function whose global minimum represents the optimal solution. The link scheduling problem can be reduced to the graph coloring problem, which can in turn be reduced to the maximum independent set problem. Simulations indicate that, for the latter problem, convex relaxation computes significantly better solutions than neural network methods based on gradient descent.

In the paper [14], we present another new neural network method called *tabu learning*, which is an adaptation of tabu search [55, 56] to neural networks. In tabu learning, the energy function associated with the neural network is changed continuously by increasing the energy in a neighborhood of the state currently being visited, thus penalizing states near those already visited. This enables the state trajectory to climb out of local minima while tending toward areas not yet visited, thus performing an efficient search of the solution space. Simulations of tabu learning applied to the traveling salesman problem and the maximum

independent set problem indicate that tabu learning computes solutions of a given cost about 300 times faster than repetitive gradient descent with random initial states.

10 ROBUST ROUTING IN NETWORKS SUBJECT TO SOPHISTICATED ATTACKS

Network survival under battle or electronic war conditions is an issue of crucial importance. The common mistake of most research done on this subject is that, to make the problems mathematically solvable, unreasonable simplifying assumptions are made about the enemy's jamming capabilities and strategies. In reality, adversaries will not do what makes the modeling task easy but what is best for them. Namely, adversaries will do whatever within their power to maximize their advantage over our forces. Another common mistake in the current electronic warfare research is to ignore the two-way dependence between the friendly and enemy actions. Under such an approach, one computes the best reaction to each possible enemy action. While such an approach is correct for dealing with an enemy with low intelligence capabilities, the enemy with more detailed knowledge of our network architectures and protocols is surely going to base his actions on the anticipated reactions that we may take. Thus, friendly and enemy actions depend on each other and cannot be dealt with in a sequential manner. One needs to treat such electronic warfare as an antagonist game, in which both the enemy's countermeasures and our counter-countermeasures are determined as part of the same optimization problem. The forthcoming paper [15] will present the results of our research in this area, and is summarized below.

Using game theory, we have developed a new method for computing a routing strategy that maximizes the worst-case end-to-end throughput under conditions that result in uncertain reduction of link capacities (e.g., jamming, atmospheric noise, friendly interference). The problem we have solved can be formulated as a game with two players. Player 1 must route R units of traffic from a source to a destination in a network with given link capacities, and Player 2 can reduce the capacity of any link by any amount, as long as the total capacity reduction does not exceed a given number. The throughput depends on the strategies of both players. Player 1 tries to maximize throughput while Player 2 tries to minimize throughput. Each player must choose a strategy without knowing that of the other player, and each player's strategy can be randomized (i.e., chosen randomly from a set of strategies).

We have shown that, because the throughput is a concave function of the routing strategy, the optimal routing strategy is pure (not randomized). We have also shown that, when all links have the same capacity, an optimal routing strategy can be found by solving the maximum flow problem for the modified network in which all link capacities are reduced to R/C, where C is the capacity of a minimum cut in the original network. When link capacities are arbitrary, this solution is approximately optimal. This says roughly that the flow should be spread over as many paths as possible, which agrees with intuition. We have also developed an algorithm for computing the optimal strategy for Player 2, which is randomized.

In addition, we have developed an efficient algorithm for computing a routing strategy that minimizes the worst-case expected end-to-end delay in a network in which each link has an unknown probability of failing, where we impose constraints on the probability assignment.

11 CONCLUSION

In this project, we have made several significant advances in the state of the art of communication networks. However, much more work is necessary. Communication networks are constantly evolving: Higher speed links are becoming available and new applications must be supported. Algorithms are needed that support both high-speed and low-speed links that support integrated services, including data, voice, and video, and that fairly and optimally allocate resources among users with diverse requirements and priorities.

Thus, the problems that must be solved are becoming more complex. Fortunately, improved computing capabilities now exist, and should be exploited. In this project, for example, we have begun to explore the ability of parallel computers (in particular, neural networks) to solve some of the difficult problems arising in communication networks. Past research has focused on solving oversimplified subproblems that are considered independently of each other; often the subproblems are highly interdependent (e.g., routing and flow control). Future research will emphasize the solution of complex problems that more accurately model the actual network, that represent combined functions (such as routing, flow control, and scheduling), and that optimize combined objectives (such as a trade-off among high throughput, low delay, and fairness) that take into account the diverse user requirements and priorities. The increased processing power of future computers, along with new computing methods, will make the solution of these complex problems possible.

BIBLIOGRAPHY

[1] R.G. Ogier and N. Shacham. A distributed algorithm for finding shortest pairs of disjoint paths. *Proc. IEEE INFOCOM*, April 1989.

[2] R.G. Ogier, V. Rutenburg, and N. Shacham. Distributed algorithms for computing shortest pairs of disjoint paths – Part I. Submitted to *IEEE Trans. Information Theory*, January 1991.

[3] R.G. Ogier, V. Rutenburg, and N. Shacham. Distributed algorithms for computing shortest pairs of disjoint paths – Part II. Submitted to *IEEE Trans. Information Theory*, January 1991.

[4] V. Rutenburg, Efficient distributed algorithms for computing shortest pairs of maximally disjoint paths in communication networks. *Proc. IEEE INFOCOM*, Miami, Florida, April 1991.

[5] R.G. Ogier and V. Rutenburg. Minimum-expected-delay alternate routing. Submitted to IEEE INFOCOM'92, June 1991.

[6] V. Rutenburg and R.G. Ogier, Fair charging policies and minimum-expected-cost routing in internets with packet loss. *Proc. IEEE INFOCOM*, Miami, Florida, April 1991.

[7] J.J. Garcia-Luna-Aceves. Loop-free internet routing and related issues. *ConneXions*, Vol. 3, No. 8, pp. 8-18, August 1989.

[8] J.J. Garcia-Luna-Aceves. A unified approach for loop-free routing using link states or distance vectors. *ACM Computer Communication Review*, Vol. 19, No. 4, pp. 212-223, September 1989.

[9] J.J. Garcia-Luna-Aceves. Loop-free routing using diffusing computations. Submitted to *IEEE Trans. Communications*, 1991.

[10] N. Shacham. Multipoint communication by hierarchically encoded data. Submitted to IEEE INFOCOM'92, June 1991.

[11] N. Shacham. Stochastic models for multihop packet radio networks. H. Takagi, ed., *Stochastic Analysis of Computer and Communication Systems*, pp. 733–765. North-Holland, 1990.

[12] V. Rutenburg and R.G. Ogier Extracting Maximum Information from Event-Driven Topology Updates. Submitted for publication.

[13] R.G. Ogier and D. Beyer. Neural network solution to the link scheduling problem using convex relaxtion. *Proc. IEEE GLOBECOM*, December 1990.

[14] D. Beyer and R.G. Ogier. Tabu learning: A neural network search method for solving nonconvex optimization problems. *Proc. International Joint Conference on Neural Networks (IJCNN)*, Seattle, Washington, July 1991.

[15] V. Rutenburg and R. Ogier, Robust Routing in Unreliable Networks. In preparation.

[16] B. Awerbuch. Complexity of network synchronization. *Journal of the Association for Computing Machinery*, 32(4):804–823, 1985.

[17] B. Awerbuch. A new distributed depth-first-search algorithm. *Information Processing Letters*, 20:147–150, 1985.

[18] A. Itai and M. Rodeh. The multi-tree approach to reliability in distributed networks. *Proc. 25th Symposium on Foundations of Computer Science*, pp. 137–147, 1984.

[19] M.L. Gardner, I.S. Loobeek, and S.N. Cohn. Type-of-service routing with loadsharing. *Proc. IEEE GLOBECOM*, Tokyo, Japan, 1987.

[20] J.W. Suurballe. Disjoint paths in a network. *Networks*, 4:125–145, 1974.

[21] D. Bertsekas and R. Gallager. *Data Networks.* Prentice-Hall, 1987.

[22] B. Awerbuch. Reducing complexities of the distributed max-flow and breadth-first-search algorithms by means of network synchronization. *Networks*, 15:425–437, 1985.

[23] D.M. Topkis. A k shortest path algorithm for adaptive routing in communication networks. *IEEE Trans. Communications*, 36(7):855–859, July 1988.

[24] L.G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Computing*, 8:410–421, 1979.

[25] D. Clark. Policy routing in internet protocols. RFC 1102, Network Information Systems Center, SRI International, Menlo Park, California, May 1989.

[26] H. W. Braun. Models of policy based routing. RFC 1104, Network Information Systems Center, SRI International, Menlo Park, California, June 1989.

[27] K. Lougheed and Y. Rekhter. A border gateway protocol. RFC 1105, Network Information Systems Center, SRI International, Menlo Park, California, June 1989.

[28] D. Estrin. Policy requirements for inter-administrative domain routing. RFC 1125, Network Information Systems Center, SRI International, Menlo Park, California, November 1989.

[29] R. E. Tarjan. *Data Structures and Network Algorithms.* Soc. Ind. Appl. Math., 1983.

[30] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. Assoc. Comput. Mach.*, pp. 1-13, 1977.

[31] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart & Winston, 1987.

[32] M. Schwartz. *Telecommunication Networks: Protocols, Modeling and Analysis*, Chapter 6. Addison-Wesley Publishing Co., Menlo Park, California, 1986.

[33] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, NJ, 1962.

[34] J.M. Jaffe and F.M. Moss. A responsive routing algorithm for computer networks. *IEEE Trans. Communications*, Vol. COM-30, No. 7, pp. 1758-1762, July 1982.

[35] K.G. Shin and M. Chen. Performance analysis of distributed routing strategies free of ping-pong-type looping. *IEEE Trans. Computers*, Vol. COMP-36, No. 2, pp. 129-137, February 1987.

[36] International Standards Organization. Intra-Domain IS-IS Routing Protocol. ISO/IEC JTC1/SC6 WG2 N323, September 1989.

[37] R. Coltun. OSPF: An Internet Routing Protocol. *ConneXions*, Vol. 3, No. 8, pp. 19-25, August 1989.

[38] E.W. Dijkstra and C.S. Scholten. Termination detection for diffusing computations. *Information Processing Letters*, Vol. 11, No. 1, pp. 1-4, August 1980.

[39] W. Zaumen and J.J. Garcia-Luna-Aceves. Dynamics of distributed shortest-path routing algorithms *Proc. ACM SIGCOMM*, Zurich, Switzerland, September 1991.

[40] P.M. Merlin and A. Segall. A failsafe distributed routing protocol. *IEEE Trans. Communications*, Vol. COM-27, No. 9, pp. 1280-1288, September 1979.

[41] E. Nussbaum. Communication network needs and technologies–A place for photonic switching? *IEEE Journal on Selected Areas in Communications*, 6(7):1036–1043, August 1988.

[42] J. S. Turner. Design of a broadcast packet switching network. *IEEE Trans. Communications*, 36(6):734–743, June 1988.

[43] J. Postel. Transmission control protocol. RFC 793, Network Information Systems Center, SRI International, Menlo Park, California, September 1981.

[44] S.E. Deering and D.R. Cheriton. Multicast routing in datagram internetworks and extended LANs. *ACM Trans. Comp. Systems*, 8(2):85–110, May 1990.

[45] C.-H. Chow. On multicast path finding algorithms. *Proc. IEEE INFOCOM*, pp. 1274–1283, Miami, Florida, April 1991.

[46] I. Gopal and J. Jaffe. Point to multipoint communication over broadcast links. *IEEE Trans. Communications*, COM-32(9), September 1984.

[47] D. Towsley and S. Mithal. A selective repeat ARQ protocol for a point to multipoint channel. *Proc. IEEE INFOCOM*, pp. 521–526, San Francisco, California, April 1987.

[48] N. Shacham and D. Towsley. Resequencing delay and buffer occupancy in selective repeat ARQ with multiple receivers. *Proc. IEEE INFOCOM*, New Orleans, Louisiana, 1988.

[49] A. Gopal, I. Gopal, and S. Kutten. Broadcast in fast networks. *Proc. IEEE INFOCOM*, pp. 338–347, San Francisco, California, June 1990.

[50] G. Karlsson and M. Vetterli. Packet video and its integration into the network architecture. *IEEE Journal on Selected Areas in Communications*, 7(5):739–751, June 1989.

[51] M. Ghanbari. Two-layer coding of video signals for VBR networks. *IEEE Journal on Selected Areas in Communications*, 7(5):771–781, June 1989.

[52] N. Shacham and P. McKenney. Packet recovery in high-speed networks using coding and buffer management. *Proc. IEEE INFOCOM*, pp. 124–131, San Francisco, California, June 1990.

[53] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biol. Cybern.*, 52:141–152, 1985.

[54] H. Szu. Fast TSP algorithm based on binary neuron output and analog neuron input using the zero-diagonal interconnect matrix and necessary and sufficient constraints of the permutation matrix. *IEEE International Conference on Neural Networks*, Vol. II, pp. 259–266, July 1988.

[55] D. de Werra and A. Hertz. *Tabu Search Techniques: A Tutorial and an Application to Neural Networks.* Ecole Polytechnique Federale de Lausanne, Deparment de Mathematiques Chaire de Recherche Operationelle, CH-1015, ORWP 89/02, 1989.

[56] M. Malek, M. Guruswamy, M. Pandya, and H. Owens. Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. Technical Report, Dept. of Electrical and Computer Engineering, The University of Texas at Austin, April 1989.